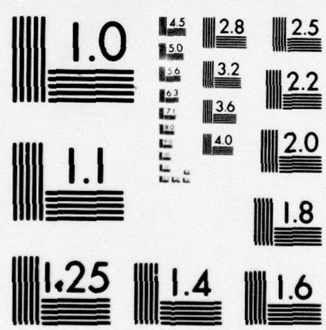
AD-A077 391 MARYLAND UNIV COLLEGE PARK COMPUTER SCIENCE CENTER F/G 9/2
PARALLEL SIGMA-ERASING ARRAY ACCEPTORS.(U)
JUL 79 A NAKAMURA AFOSR-77-3271
UNCLASSIFIED TR-784 AFOSR-TR-79-1163 NL

1 OF 1
ADA
077391

END
DATE
FILMED
12-79
DDC



MICROCOPY RESOLUTION TEST CHART

AFOSR-TR- 79-1163

(10)

LEVEL

AD A077391

DDC FILE COPY

UNIVERSITY OF MARYLAND

COMPUTER SCIENCE CENTER

COLLEGE PARK, MARYLAND

20742

79 11 27 091

TR-784 July 1979
AFOSR-77-3271

PARALLEL Σ-ERASING ARRAY ACCEPTORS

Akira Nakamura

Department of Applied Mathematics
Hiroshima University (Japan)
Hiroshima, Japan

and

Computer Science Center
University of Maryland
College Park, MD 20742

See 1473 in back.

ABSTRACT

A parallel Σ-erasing array acceptor (PΣ-EAA) is introduced. It is proved that the class accepted by PΣ-EAA's is exactly the context-free array languages.

The support of the U.S. Air Force Office of Scientific Research under Grant AFOSR-77-3271 is gratefully acknowledged, as is the help of Kathryn Riley in preparing this paper.

1. Introduction

In [1], Rosenfeld introduced isotonic array grammars, and in [2] Cook and Wang presented a Chomsky hierarchy of isotonic array grammars and languages. Acceptors for type 0 and type 1 array languages were given in Milgram and Rosenfeld [3] and an acceptor for type 3 array languages was given in [2]. But an acceptor for type 2 or context-free languages remains as an open problem.

Cook and Wang's finite state array acceptors (FSAA's) for type 3 array languages are considered as erasing array acceptors in the following sense: At each step, an FSAA M is in some state q scanning an input symbol a. In a given move, M marks the scanned array symbol with ¢ (the input symbol is erased by changing it to a symbol with ¢), changes state and moves one square left (L), right (R), up (U), or down (D). An array A is accepted iff M, beginning in its initial state scanning some array cell, scans and marks each non-# square of the A with a ¢ and ends in one of its final states. In this definition of acceptability, there is the problem that M itself never knows whether every non-# cell is marked with ¢.

In this note, we introduce a kind of erasing array acceptor similar to the above mentioned one, which acts in parallel and erases input symbols according to given rules. Further, the acceptor is able to traverse the area of input symbols after a cell falls into a special state. This ability is provided in order to avoid the above-mentioned problem. This automaton can be considered as a restricted cellular array acceptor; it is called a <u>parallel Σ-erasing array acceptor</u> (PΣ-EAA).

The purpose of this note is to solve the open problem concerning the context-free array languages proposed in [2]. We prove that the class accepted by PΣ-EAA's is exactly the context-free array languages. A key idea in the proof is based on a definition of a nondeterministic PΣ-EAA such that it is able to retrace backward derivations in an isotonic context-free array grammar.

In this note, we assume that the reader is familiar with isotonic array grammars, languages, and their hierarchy given in [1], [2] and also with the fundamental definitions of cellular array acceptors as in [4].

2. Definitions

We first give some notation and definitions about PΣ-EAA's. In this note, we assume that input arrays are surrounded by the edge symbol #.

<u>Definition 2.1</u> A parallel Σ-erasing array acceptor (PΣ-EAA) is a two-dimensional cellular acceptor satisfying the following (i)-(vi):

(i) For a set Σ of input states, a set Q of non-input states, and a set T of traversal states, $\Sigma \cap Q = \phi$, $\Sigma \cap T = \phi$, $Q \cap T = \phi$.

(ii) For each $A \in Q$, A^L, A^R, A^U, and A^D are also in Q.

(iii) $\not{b} \in Q$ is a special "dead" blank state.

(iv) S^t, $S \in Q$ are the quasi-accepting state and the accepting state, respectively.

(v) $\overline{S} \in Q$ is the "broken" state.

(vi) δ is the transition function, i.e. $\delta:(\Sigma \cup Q \cup T)^5 \to 2^{(\Sigma \cup Q \cup T)}$

In this note, we consider exclusively nondeterministic acceptors. In Definition 2.1, $\delta(X,Y,Z,U,V) \ni W$ means that one of the values of δ over $\begin{array}{|c|c|c|} \hline & Y & \\ \hline X & V & Z \\ \hline & U & \\ \hline \end{array}$ is $\begin{array}{|c|c|c|} \hline & * & \\ \hline * & W & * \\ \hline & * & \\ \hline \end{array}$. That is, the first 4 arguments of δ are occupied by the 4 neighbors in the indicated positions of the last argument. For intuitive understanding, $\delta(X,Y,Z,U,V)$ is sometimes denoted by $\delta\begin{pmatrix} & Y & \\ X & V & Z \\ & U & \end{pmatrix}$.

δ is defined as a mapping satisfying the following (1)-(10):

(1) $\delta(X,Y,Z,U,\not{b})$ is always $\not{b}$, except in the case of traversal mentioned in (9) below. This means that $\not{b}$ is the special dead blank state.

(2) $\delta(X,Y,Z,U,A) \ni A^H$, where $H \in \{\mathcal{L},\mathcal{R},\mathcal{U},\mathcal{D}\}$, in cases not contradicting (5) and (6) below.

(3) For each $H \in \{\mathcal{L},\mathcal{R},\mathcal{U},\mathcal{D}\}$ and $A \in Q$, $\delta(X,Y,Z,U,A^H)$ is always $\not{b}$. Conditions (1) and (3) correspond to "erasing".

(4) For each input state a, the first 4 variables X,Y,Z,U of $\delta(X,Y,Z,U,a)$ are dummy.

(5) $\delta(A\ \overset{B}{\underset{D^{\mathcal{U}}}{E}}\ C)$ depends on E and D only. In the other cases, the situation is the same, i.e., for example $\delta(A\ \overset{B}{\underset{D}{E}}\ C^{\mathcal{L}})$ depends on E and C only, $\delta(A^{\mathcal{R}}\ \overset{B}{\underset{D}{E}}\ C)$ depends on E and A only, and so on.

(6) In case of a collision in (5), the acceptor goes into the broken state $\overline{S}$, i.e., for example, $\delta(A\ \overset{B}{\underset{D^{\mathcal{U}}}{E}}\ C^{\mathcal{L}}) = \overline{S}$, $\delta(A^{\mathcal{R}}\ \overset{B^{\mathcal{R}}}{\underset{D}{E}}\ C^{\mathcal{L}}) = \overline{S}$, and so on.

(7) For the broken state $\overline{S}$, $\delta(X,Y,Z,U,\overline{S}) = \overline{S}$.

(8) $\delta(X\ \overset{Y}{\underset{U}{E}}\ Z)$ takes a value different from E only when $X = A^{\mathcal{R}}$, $Y = B^{\mathcal{D}}$, $Z = C^{\mathcal{L}}$, or $U = D^{\mathcal{U}}$ for some A,B,C,D. This corresponds to the "context-free" property.

(9) When a cell falls into the quasi-accepting state S^t, δ is defined to begin traversal of the input. The states of the set T are used for this traversal.

(10) In the process of traversal, a cell changes to the broken state when two states of $\Sigma \cup Q$ are in the input area;

otherwise it falls into the accepting state at the starting cell of S^t after finishing the traversal. The detailed definition of δ to satisfy (10) is complicated and will not be given here.

3. Main theorem

We prove that the class accepted by PΣ-EAA's is exactly the isotonic context-free array languages. Let G be an isotonic context-free array grammar. L(G) represents the language generated by this grammar. Also, let M be a PΣ-EAA. T(M) means the set accepted by this acceptor M.

Lemma 3.1 Any context-free array language is accepted by a PΣ-EAA.

Proof:

Let G be a type 2 grammar G=(N,T,#,P,S) and let M be a PΣ-EAA. We prove that $\forall G \exists M (L(G)=T(M))$.

Any isotonic context-free array language (ICFAL) is generated by the following normal form of rewriting ([2]):

$$\#A \to BC, \quad A\# \to BC, \quad \begin{array}{c}\#\\A\end{array} \to \begin{array}{c}B,\\C\end{array} \quad \begin{array}{c}A\\\#\end{array} \to \begin{array}{c}B,\\C\end{array} \quad A \to a.$$

Therefore, an array generated by the normal form is accepted by a PΣ-EAA which retraces backward rewriting rules. That is, δ of this PΣ-EAA is defined as follows:

$$\delta(.,.,.,.,a) \ni A \qquad \text{if } A \to a \text{ is in } P,$$

$$\delta\left(\begin{array}{ccc} & \cdot & \\ \cdot & B & C^{L} \\ & \cdot & \end{array}\right) \ni A \qquad \text{if } A\# \to BC \text{ is in } P,$$

$$\delta\left(\begin{array}{ccc} & \cdot & \\ B^{R} & C & \cdot \\ & \cdot & \end{array}\right) \ni A \qquad \text{if } \#A \to BC \text{ is in } P,$$

$$\delta\left(\begin{array}{ccc} & & \\ \cdot & B & \cdot \\ & C^{U} & \end{array}\right) \ni A \qquad \text{if } \begin{array}{c}A\\\#\end{array} \to \begin{array}{c}B\\C\end{array} \text{ is in } P,$$

$$\delta\left(\begin{array}{ccc} & B^{D} & \\ \cdot & C & \cdot \\ & \cdot & \end{array}\right) \ni A \qquad \text{if } \begin{array}{c}\#\\A\end{array} \to \begin{array}{c}B\\C\end{array} \text{ is in } P.$$

In this definition, if a symbol A in the production is S, then the corresponding A for δ is S^t. It is easily seen that an input array ends in the accepting state S at some square and each non-# cell of the array goes into $\not{b}$ iff the input array is generated by G.//

<u>Lemma 3.2</u> The set accepted by a PΣ-EAA is a context-free array language.

<u>Proof</u>:

The proof is similar to Lemma 3.1. We prove that $\forall M \exists G(L(G)=T(M))$. We define an isotonic context-free array grammar $G=(N,T,\#,P,S)$ from a given PΣ-EAA M as follows:

$$A \rightarrow a \qquad \text{if } \delta(.,.,.,.,a) \ni A$$

$$A\# \rightarrow BC \qquad \text{if } \delta\left(\begin{array}{ccc} & \cdot & \\ \cdot & B & C^L \\ & \cdot & \end{array}\right) \ni A$$

$$\#A \rightarrow BC \qquad \text{if } \delta\left(\begin{array}{ccc} & \cdot & \\ B^R & C & \cdot \\ & \cdot & \end{array}\right) \ni A$$

$$\begin{array}{c} A \\ \# \end{array} \rightarrow \begin{array}{c} B \\ C \end{array} \qquad \text{if } \delta\left(\begin{array}{ccc} & \cdot & \\ \cdot & B & \cdot \\ & C^U & \end{array}\right) \ni A$$

$$\begin{array}{c} \# \\ A \end{array} \rightarrow \begin{array}{c} B \\ C \end{array} \qquad \text{if } \delta\left(\begin{array}{ccc} & B^D & \\ \cdot & C & \cdot \\ & \cdot & \end{array}\right) \ni A$$

$$S \rightarrow S^t$$

For the other cases of δ, the corresponding rewriting rules are not given. It can be seen without difficulty that an input array is generated by this isotonic context-free array grammar G iff it is accepted by M.//

Theorem 3.3 The class accepted by PΣ-EAA's is exactly the isotonic context-free array languages.

Proof:

This is obtained from Lemmas 3.1 and 3.2.//

4. Remark

In this note, we have considered parallel Σ-erasing array acceptors for the isotonic context-free array languages. We could also define sequential acceptors which simulate PΣ-EAA's; but it seems that these acceptors are somewhat more complicated.

Acknowledgement

The author would like to thank Professor Azriel Rosenfeld for arranging a very enjoyable four week stay at the Computer Science Center, July 1979, and for valuable comments. He also thanks Dr. T. Ae of Hiroshima University for his useful suggestions.

References

[1] A. Rosenfeld: Isotonic grammars, parallel grammars and picture grammars, in Machine Intelligence 6 (D. Michie and B. Meltzer, Eds.), 281-294, Univ. of Edinburgh Press, Edinburgh, Scotland, 1971.

[2] C. R. Cook and P. S. Wang: A Chomsky hierarchy of isotonic array grammars and languages, Computer Graphics and Image Processing 8, 144-152, 1978.

[3] D. L. Milgram and A. Rosenfeld: Array automata and array grammars, Proc. IFIP Congress 71, booklet TA-2, 166-173, North-Holland, Amsterdam, 1971.

[4] A. Rosenfeld: Picture Languages, Academic Press, New York, 1979.

Unclassified　　　　　　　　　　　　19 TR-79-1163

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 18 AFOSR	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 PARALLEL Σ-ERASING ARRAY ACCEPTORS. sigma		5. TYPE OF REPORT & PERIOD COVERED 9 Technical Rept.
		6. PERFORMING ORG. REPORT NUMBER 14 TR-784
7. AUTHOR(s) 10 Akira/Nakamura		8. CONTRACT OR GRANT NUMBER(s) 15 AFOSR-77-3271
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Science Center University of Maryland College Park, MD 20742		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 17 A2　16 2304/A2 61102F
11. CONTROLLING OFFICE NAME AND ADDRESS Math. & Info. Sciences, AFOSR/NM Bolling AFB Wash., DC 20332		12. REPORT DATE 11 Jul[redacted]79
		13. NUMBER OF PAGES 12 13
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Automata
Formal languages

Sigma

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A parallel Σ-erasing array acceptor (PΣ-EAA) is introduced. It is proved that the class accepted by PΣ-EAA's is exactly the context-free array languages.

DD FORM 1 JAN 73 1473　EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

4A3 A49